

**Arc64**

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i> Arc64		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		May 28, 2025	

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>Arc64</b>	<b>1</b>
1.1	Arc64 documentation (14th Apr.98)	1
1.2	Disclaimer	1
1.3	About Arc64	1
1.4	Requirements	2
1.5	How to use Arc64	2
1.6	What is still left to do?	4
1.7	Thanx to...	4
1.8	Bugs	4
1.9	History	4
1.10	How to contact the author	6
1.11	P00	6
1.12	T64	6
1.13	LYNX	7
1.14	Arkive	8
1.15	D64	8
1.16	ZipCode	9
1.17	Convert to D64	10

# Chapter 1

## Arc64

### 1.1 Arc64 documentation (14th Apr.98)

```
Arc64 V1.9
© 1996-98 by John 'Graham' Selck
```

```
Disclaimer
About
Requirements
How to use
Future
Thanx to...
Known Bugs
History
Contact
```

### 1.2 Disclaimer

Arc64 is written and copyright © 1996-98 by John Selck.

This program is freely distributable unless no changes are made to the archive.

The author is not liable for any damage/problems/loss of data this program might directly or indirectly cause.

No parts of this program may be altered or resourced in any way.

### 1.3 About Arc64

Arc64 is a program designed to give you fast access to files stored in c64 archives and 1541 disk images.

It was mainly written because 'UNd64' (© by Jess Sosnoski) didn't support ZipCode archives.

---

(Anyway, now it does...)

However, Arc64 is meant as an alternative towards UNd64.

In comparision Arc64 has the following advantages:

- ZipCode support
- less textoutput
- c64-like directory output
- supports T64 V1.1
- supports Arkive
- conversion from file to (existing) disk archives

## 1.4 Requirements

To use Arc64 you need:

- an Amiga

Arc64 has been tested by me on following configurations:

- A500, 68000, 0.5MB Fast, Kick 1.3 (Yes, finally it works with 1.3!)
- A1200, 68030, 32MB Fast, Kick 3.0
- A4000, 68060, 64MB Fast, Kick 3.1

## 1.5 How to use Arc64

Arc64 is used via CLI.

It supports P00, T64, Lynx, Ark, D64 and ZipCode.

Arc64 {command} {archive name} [{archive name}]

Commands are: l = List directory/archived files  
x = Extract files from archive  
c = Convert ZipCode to D64 / D64 to ZipCode  
Convert to D64  
s = Store a file to a D64 archive  
v = validate a D64/ZipCode diskimage  
z = zero-fill a D64/ZipCode diskimage  
a = Add directory to logfile

Example:

Arc64 x HD2:D64/vendetta.d64

...will extract all files from the d64 image into the actual directory.

Arc64 c DH1:1!Demo\_archive

...will convert the ZipCode archive "Demo\_archive" into a plain

---

d64 archive. (This also works the other way round...)

```
Arc64 c Test/4kntscm.lnx
```

... will extract all files into a D64 archive which will be named "4kntscm.d64".

```
Arc64 c Test/4kntscm.lnx ugh.d64
```

... will extract all files from the archive "4kntscm.lnx" into the "ugh.d64" disk archive. If the disk archive doesn't exist, a new one will be created. Be aware that Arc64 doesn't check the BAM for being valid!

```
Arc64 s Test/4kntscm.lnx ugh.d64
```

... will NOT extract the LNX archive but will store it as raw data. This works with any file!

```
Arc64 v d0138a.d64
```

... will rescan the files on the disk and create a new BAM out of it. This will also work with a corrupted file structure. Be aware that this one will not notice 'hidden' data on the disk.

```
Arc64 z d0138a.d64
```

... this one will fill all empty blocks with zeros. I recommend you to do a validate before this one. (For example, some directory tools on c64 do not update the BAM, this can cause problems!)

I recommend that you use Arc64 from DirOpus or something similar.

If you wonder about the 'a' command, i have to say that i included this one for myself.

```
Arc64 a CD0:Arc/xxx.d64
```

will basically do the same as

```
Arc64 l CD0:Arc/xxx.d64 >Arc64_Log
```

with the difference that the name of the actual archive will still be written onto the screen.

(The logfile will be created on the actual path as 'Arc64\_Log'.)

Errors which may appear while listing:

Illegal Link

- Same as Illegal Track/Sector, only for the directory structure.

Looped Directory

- This happens quite often since people always used to play around

---

with the directory.

## 1.6 What is still left to do?

- ??? any idea ???

## 1.7 Thanx to...

Thanx to...

Timo (Deadbeat)

... the master of all bugreports!!!

Andreas (CountZero)

... for his major bugreports and a T64 file Arc64 couldn't handle

Geert Verschueren (Sorex/WOW)

... for that strange Lynx file with that 0 blocks USR files

Paul Gardner-Stephen (Highlander/FLT)

... for his encouragement giving me information on ZipCode  
(although i found out myself...)

Jess Sosnoski

... for UNd64

## 1.8 Bugs

Known Bugs/Problems

- Disk Full requester results in a crash when cancel is pressed!

For bugreports, press this one...

## 1.9 History

History

1.9 - Introduced a new command: 'z' fills all free blocks of a d64 or ZipCode image with zeros.

- Now validate also works with ZipCode archives.

- Added the possibility to change the header of a new created d64

---

- image via an environment variable called "Arc64.header".
  - For ZipCode archives it is not longer necessary to have the "1!" at the beginning, you can use "Arc64 x zipname" instead of "Arc64 x 1!zipname".
  - BF: Filenames with ISO characters caused the rest of the name to be cropped.
  - BF: Several accesses to odd addresses have been removed, so the 68000 compability is back again.
  - BF: The filename to D64/ZipCode header conversion did not work 100% together with the 'store' command.
  - BF: (kind of...) The ASL requester for the 'store' command only had shown the supported archives in a directory. Since this command affects all kinds of files, this has been changed.
- 1.8 - Arc64 now removes ".Z64" suffixes from ZipCoded archives when converting them to D64. No more ugly ".z64.d64" anymore...
- When listing a directory, there was either "<" or " " after the filetype. " " will now be skipped, so files containing lots of directories will be a few bytes shorter now...
  - BF: archives with a file size of 0 bytes crashed Arc64.
- 1.7 - Added "store" function which i missed many times when using Arc64
- BF: fixed a small bug in the text output of the "c" command
  - Arc64 does not change blanks into '\_' anymore
- 1.6 - BF: sizes when listing older T64 archives were wrong in some cases  
BF: when a directory entry was linked to an illegal track or sector, all following files were skipped  
Now files will not be extracted if they link up to the directory
- 1.5 - now it is possible to extract file archives into an existing D64 image.  
BF: selecting files with an ASL requester caused a crash.
- 1.4b- BF: the list command didn't work for D64 files in 1.4! This was the result of copy-paste in the source...
- 1.4 - added possibility to directly extract file archives into D64 images.  
added validate function for D64 archives. Well, also ZipCode validate follows but i need to recode the Zip-saveroutines.  
BF: the check for existing files in a directory corrupted the stack (this was a REAL ugly one!)
- 1.3 - ASL requester when filename is left out.  
BF: a certain block link in a D64/ZipCode archive caused crashes  
added userbreak on Ctrl-C
- 1.2 - Arkive support  
BF: the ListDir for D64/ZipCode didn't show the blocks free value.  
BF: filenames which were followed by a space sometimes caused crashes.  
Support for another strange T64 with slightly different header.
- 1.1 - T64 support (1.0 & 1.1)  
P00 support  
BF: saveroutine only saved 'ok'-files. fixed.  
The Lynx routines now notice a broken Lynx archive, same goes to T64.
-



1.0 - Initial Release

## 1.10 How to contact the author

Contact me at:

John Selck (Graham/Oxyron)  
Suederholz 13  
24885 Sieverstedt  
Germany

E-Mail: j.selck@flensburg.netsurf.de

## 1.11 P00

P00 archives are file archives containing only one file.  
They were introduced by PC64, a C64 emulator for MS-DOS.

\*\*\* Format of P00 list output:

Archive: battlflld.p00

67 "BATTLEFIELD" PRG

\*\*\* Format of T64 extract output:

Extracting from: battlflld.p00

battlefield ok

\*\*\* Possible errors:

ok No error.

Skipped... Another file with the same filename is already  
on disk.

## 1.12 T64

T64 archives are file archives containing one or more files.  
The T64 format was introduced by C64S, a C64 emulator for MS-DOS.

There are currently two versions of the T64 format.  
The first (1.0) supports only single files and the second (1.1)  
supports multiple files. Both are supported by Arc64.

\*\*\* Format of T64 list output:

Archive: gt021.t64

---

```
"GAME TAPE #021          " <- Tape Header
1    "LOADER             " PRG <- Complete file
157  "MAIN.DAT           "*PRG <- Incomplete file
33   "E000               "*DEL <- The archive doesn't contain the file
```

\*\*\* Format of T64 extract output:

Extracting from: gt021.t64

```
loader_____ ok
main.dat_____ Size error!
e000_____ Skipped...
```

\*\*\* Possible errors:

```
ok    No error.
```

Size error! Either the archive is too small  
or the file length not indicated correctly.  
However, there are a lot of T64 files which  
contain only 1 file and have the header from  
the original C64S demo tape image.

Skipped... The file is not within the archives range or  
another file with the same filename is already  
on disk.

## 1.13 LYNX

Lynx archives are file archives containing one or more files.

\*\*\* Format of Lynx list output:

Archive: iffconverternv1.lnx

```
0    "-----" USR
0    "-      1995      -" USR
0    "-----" USR
13   "IFFCONV V1.0/MSP" PRG
17   "IFFCONV DOCS/MSP" PRG
108  "TESTPICTURE.LBM" *SEQ <- Incomplete file
```

\*\*\* Format of Lynx extract output:

Extracting from: iffconverternv1.lnx

```
----- Skipped...
-      1995      - Skipped...
----- Skipped...
iffconv_v1.0/msp ok
iffconv docs/msp ok
testpicture.lbm Size error!
```

\*\*\* Possible errors:

ok      No error.

Size error! Either the archive is too small  
or the file length not indicated correctly.

Skipped... The file is not within the archives range or  
the file has a size of 0 blocks or another  
file with the same filename is already on disk.

## 1.14 Arkive

Ark archives are file archives containing one or more files.

\*\*\* Format of Ark list output:

Archive: iffconverternv1.ark

```
0      "-----" USR
0      "-      1995      -" USR
0      "-----" USR
13     "IFFCONV V1.0/MSP" PRG
17     "IFFCONV DOCS/MSP" PRG
108    "TESTPICTURE.LBM" *SEQ
```

\*\*\* Format of Ark extract output:

Extracting from: iffconverternv1.ark

```
----- Skipped...
-      1995      - Skipped...
----- Skipped...
iffconv_v1.0/msp ok
iffconv docs/msp ok
testpicture.lbm Size error!
```

\*\*\* Possible errors:

ok      No error.

Size error! Either the archive is too small  
or the file length not indicated correctly.

Skipped... The file is not within the archives range or  
the file has a size of 0 blocks or another  
file with the same filename is already on disk.

## 1.15 D64

D64 archives are complete 1541 images.

The D64 format was introduced by C64S, a C64 emulator for MS-DOS.

---

\*\*\* Format of D64 list output:

Archive: Utils1.d64

```
0 "KICKIN' ASS IN " 96 2A <- Disk header & ID
12 "TURBOCOPY 1.0" PRG <- File
166 "URIDIUM++ /DCS" PRG< <- Write protected file
108 "TOOLBOX V1.0/UCC"*PRG <- Unclosed file
48 "DEFENDER" PRG
4004 BLOCKS FREE
```

\*\*\* Format of D64 extract output:

Extracting from: Utils1.d64

```
turbocopy_1.0      ok
uridium++____\dcs ok
toolbox_v1.0\ucc Illegal Track/Sector!
defender    Looped filestructure!
```

\*\*\* Possible errors:

```
ok      No error.
```

Illegal Track/Sector! A blockpointer in the file points to a  
block out of the range of a 1541 disk.

Looped filestructure! A blockpointer points to a block earlier in  
the file.

Corrupted EOF! The last block of a file contains an invalid  
bytesize.

Skipped... A file with the same name is already on disk.

## 1.16 ZipCode

ZipCode archives are complete 1541 images.

\*\*\* Format of ZipCode list output:

Archive: 1!Utils1

```
0 "KICKIN' ASS IN " 96 2A <- Disk header & ID
12 "TURBOCOPY 1.0" PRG <- File
166 "URIDIUM++ /DCS" PRG< <- Write protected file
108 "TOOLBOX V1.0/UCC"*PRG <- Unclosed file
48 "DEFENDER" PRG
4004 BLOCKS FREE
```

\*\*\* Format of ZipCode extract output:

Extracting from: 1!Utils1

```
turbocopy_1.0      ok
```

---

```
uridium++___\dcs ok
toolbox_v1.0\ucc Illegal Track/Sector!
defender   Looped filestructure!
```

\*\*\* Possible errors:

```
ok          No error.
```

Illegal Track/Sector! A blockpointer in the file points to a  
block out of the range of a 1541 disk.

Looped filestructure! A blockpointer points to a block earlier in  
the file.

Corrupted EOF! The last block of a file contains an invalid  
bytesize.

Skipped... A file with the same name is already on disk.

## 1.17 Convert to D64

Conversion from file archive formats to D64 disk images:

This allows you to extract all files from P00, T64, LYNX and ARK  
archieves into a D64 image.

Finally you can get rid of these ugly formats :)

\*\*\* Possible errors:

```
ok          No error.
```

Disk full! The file does not fit into the D64 image anymore.

Directory full! There are no more empty slots for filenames. This  
should hardly happen.

... plus the errors which appear while extracting.

If files are shrunk (i.e. Size error) the files will be stored until  
the error occurs. The file type will be changed to 'unclosed' ("\*").